



Prisma

<https://cybersoft.edu.vn/>

CYBERSOFT

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



Mục Lục

01 Giới thiệu chung

02 Kết quả đạt được

03 Cài đặt

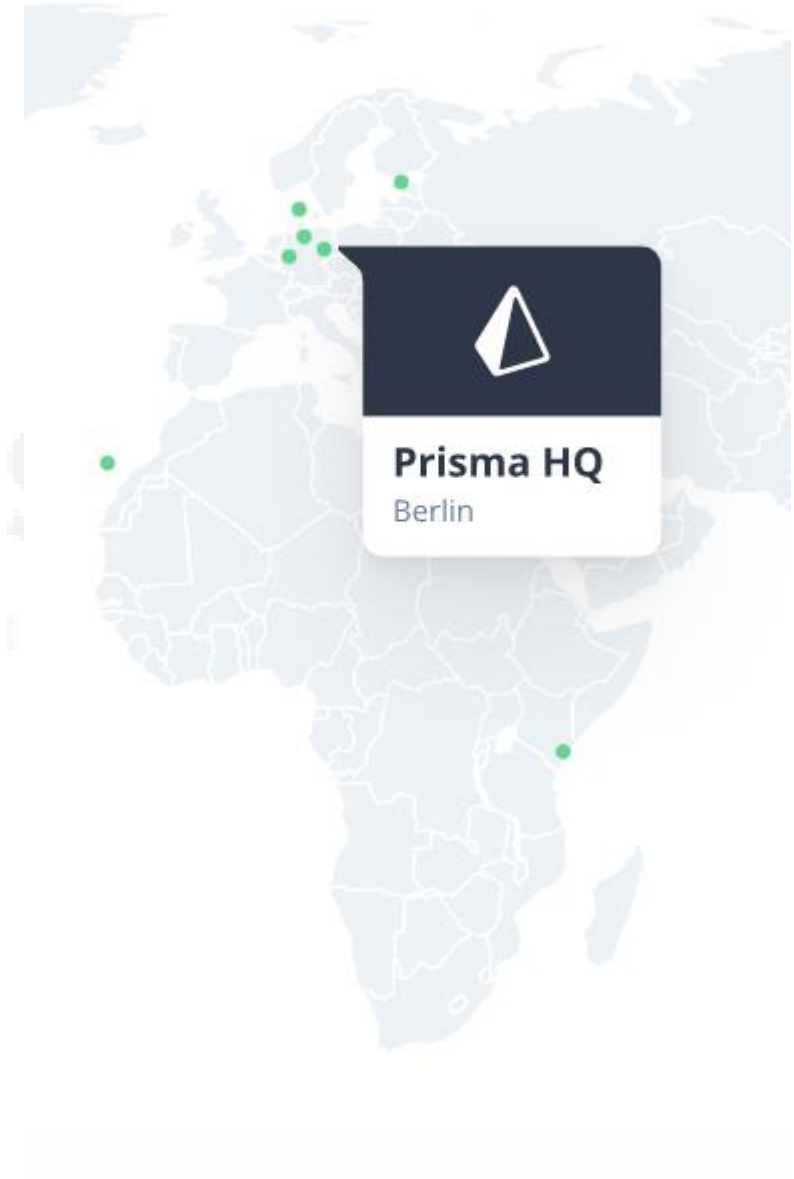
04 Migrates

05 Querying database

06 Thực hành

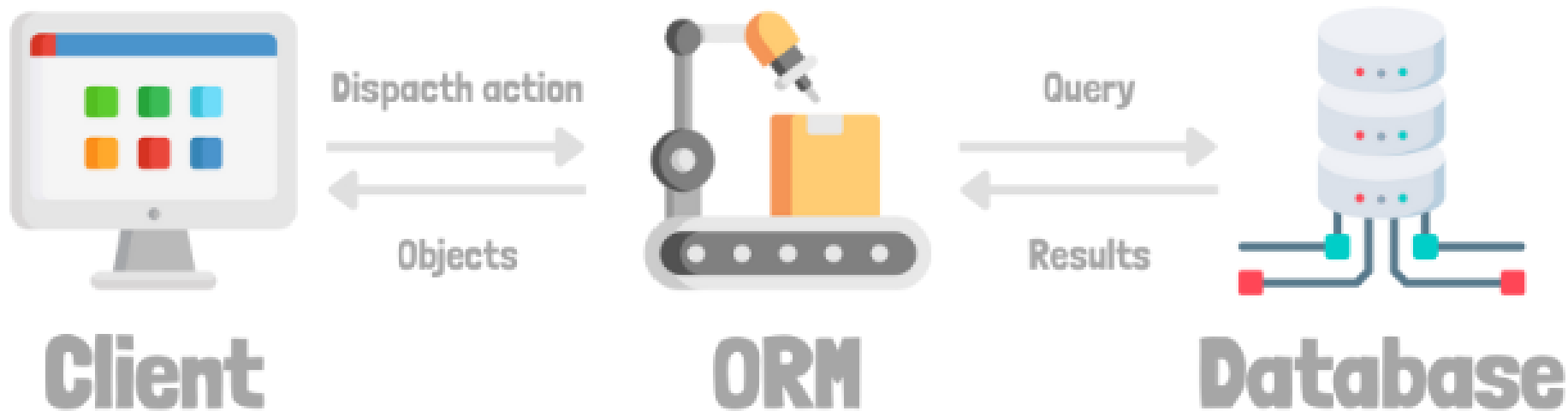
Why Prisma ?

- Sử dụng cơ chế ORM tương tự như **Sequelize**
- Tránh thao tác trực tiếp với tầng database
=> không cần viết lệnh SQL dài dòng
- Cú pháp đơn giản dễ cài đặt, sử dụng
- Hỗ trợ xây dựng **code first** lẫn **database first**



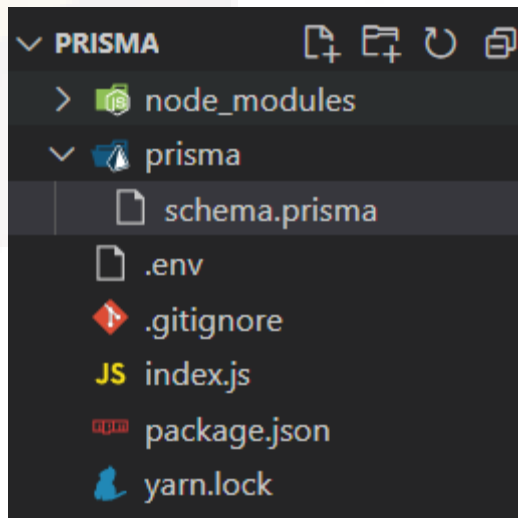
Kết quả đạt được

Thao tác với database lấy dữ liệu nhanh, đơn giản hơn



Cài đặt

- Trang chủ: <https://www.prisma.io/>
- Thêm thư viện: **yarn add prisma @prisma/client**
- Structure: **yarn prisma init**



Cài đặt: Kết nối database

- Tham khảo: <https://www.prisma.io/docs/concepts/database-connectors/mysql>
- Truy cập file **.env**

CONNECTOR	USER:PASSWORD@HOST:PORT	DATABASE	KEY1=VALUE&KEY2=VALUE
mysql	:// janedoe:mypassword@localhost:5432	/ mydb	? connection_limit=5
Protocol	Base URL	Path	Arguments

schema.prisma

Datatype (kiểu dữ liệu):

- String
- Int, BigInt
- Float, Decimal
- Boolean
- DateTime

```
JS index.js schema.prisma
prisma > schema.prisma
4
5 //CRUD
6 generator client {
7   provider = "prisma-client-js"
8 }
9
10
11 //connect database
12 datasource db {
13   provider = "mysql"
14   url      = env("DATABASE_URL")
15 }
16
17
18 //lệnh tạo bảng
19 model User {
20   user_id      Int      @id @default(autoincrement())
21   email        String
22   user_name    String?
23   type_id      Int
24   type_user    Type_User @relation(fields: [type_id], references: [id])
25 }
26
27 model Type_User {
28   id          Int      @id @default(autoincrement())
29   type_name    String?
30   user        User[]
31 }
32
33
34 //yarn add @prisma/client
35 //yarn prisma db push
36 // yarn prisma migrate dev
```

schema.prisma

Thuộc tính thêm như sql:

- @id: Khóa chính
- @@id([id1 , id2])
- @default(autoincrement())
- @unique
- @relation(fields: [] , references: [])
- Tạo field danh sách: **[User]**
- Cho phép trống: **String?**

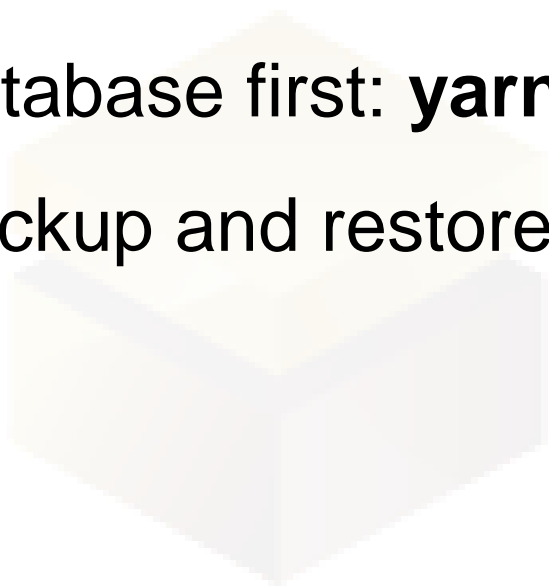
```
model User {
  id      Int      @id @default(autoincrement())
  posts   Post[]
}

model Post {
  id        Int      @id @default(autoincrement())
  author    User      @relation(fields: [authorId], references: [id])
  authorId  Int
}
```

```
const getUser = await prisma.user.findMany({
  include: {
    type_User: true
  }
});
```


Migrates

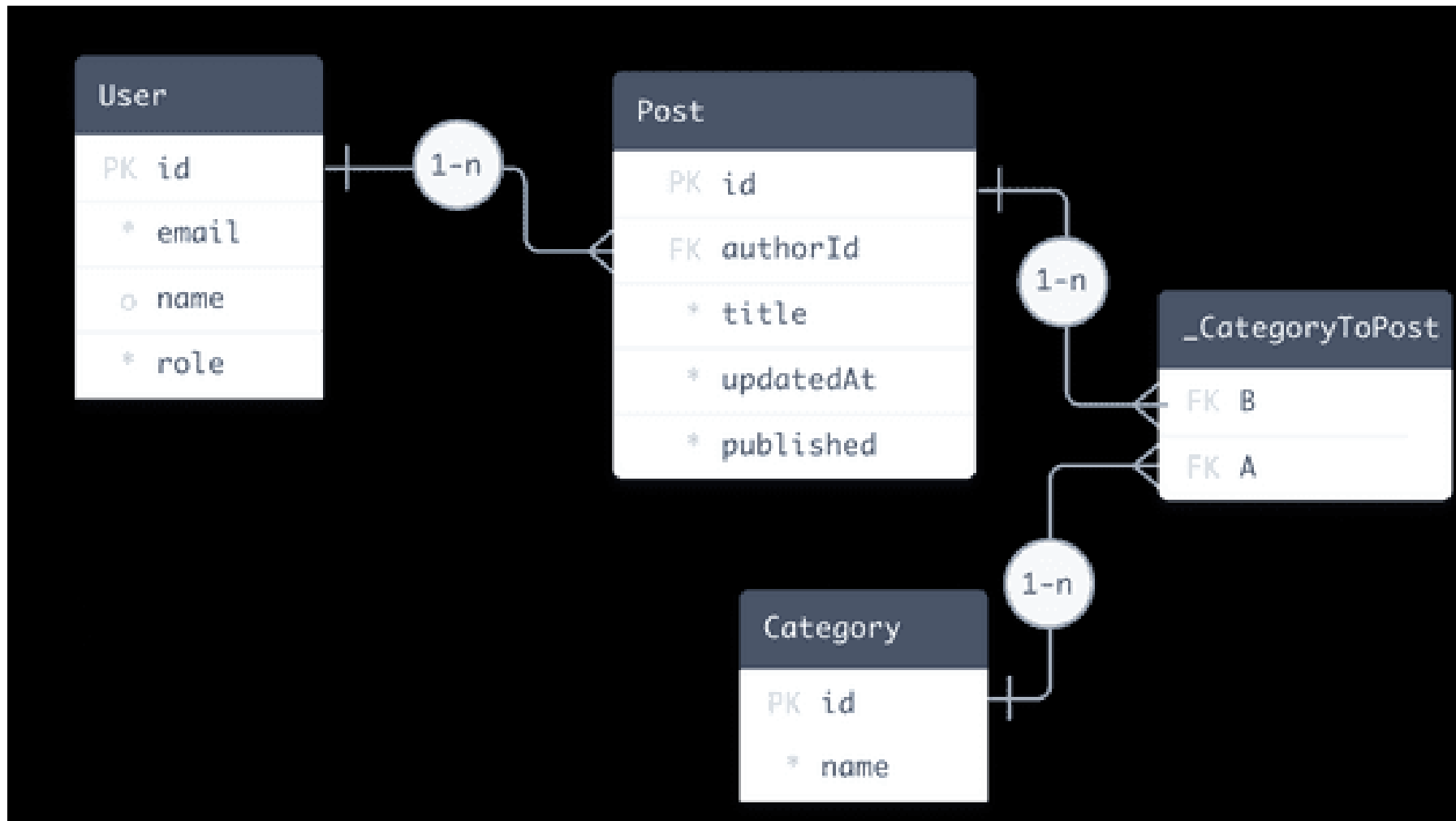
- Code first: **yarn prisma db push**
- Database first: **yarn prisma db pull**
- Backup and restore: **yarn prisma migrate dev**



CYBERSOFT
ĐÀO TẠO CHUYÊN GIA LẬP TRÌNH

Thực hành

Tạo theo Code first, đặt tên database là db_clone



Querying database

- C: `prisma.user.create()`
- R: `prisma.user.findMany()`
- U: `prisma.user.update()`
- D: `prisma.user.delete()`
- Tham khảo:
<https://www.prisma.io/docs/concepts/components/prisma-client/crud>
- `yarn prisma generate`

```
JS index.js  •  schema.prisma  .env

JS index.js > app.get("/api") callback
1  const express = require('express');
2  const app = express();
3  app.use(express.json());
4  const { PrismaClient } = require('@prisma/client')
5
6  const prisma = new PrismaClient()
7
8  app.listen(8080)
9
10 app.get("/api", async (req, res) => {
11
12     const getUser = await prisma.type_User.findMany({where:{
13         type_name:{
14             contains:"abc"
15         }
16     }
17 });
18
19 res.send(getUser)
20 })
```